

### **Remarks**

This REPLY is in response to the Office Action mailed February 20, 2008, and an interview with Examiner Barbara Burgess on August 12, 2008. The fee for addition of new claims is enclosed herewith. A Petition for Extension of Time is enclosed herewith, together with the appropriate fee.

Applicant wishes to thank Examiner Burgess for the courtesy of an interview with Karl Kenna and Mark Smit on August 12, 2008, during the course of which interview the participants generally discussed the pending Application and the cited references. No claim amendments were proposed during the interview.

#### **I. Summary of Examiner's Rejections**

Prior to the Office Action mailed February 20, 2008, Claims 1-15, 17, 24-27 and 29-60 were pending in the Application. In the Office Action, Claim 52 was objected to for informalities. Claims 52-53 were rejected 35 U.S.C. 102(e) as being anticipated by Yano et al. (U.S. Patent Publication No. 2006/0184546, hereinafter Yano). Claims 1-15, 17, 24-27, 29-51 and 54-60 were rejected under 35 U.S.C. 103(a) as being unpatentable over Yano in view of Gershman et al. (U.S. Patent No. 6,356,905, hereinafter Gershman).

#### **II. Summary of Applicant's Amendment**

The present Reply amends Claims 1, 6, 24, 29, 52 and 53; cancels Claims 61-62; and adds new Claims 63-64, as shown above, leaving for the Examiner's present consideration Claims 1-15, 17, 24-27 and 29-60 and 63-64.

#### **III. Claim Objections**

In the Office Action mailed February 20, 2008, Claim 52 was objected to for informalities. Accordingly, Claims 52 and 53 have been amended as shown above to address only these informalities. Reconsideration is respectfully requested.

#### **IV. Claim Rejections under 35 U.S.C. 102(e)**

In the Office Action mailed February 20, 2008, Claims 52-53 were rejected 35 U.S.C. 102(e) as being anticipated by Yano (U.S. Patent Publication No. 2006/0184546).

## Claim 52

Applicant respectfully traverses the rejection of Claim 52, which currently defines:

52. *A system for suggesting data as a response to client requests, comprising:*  
*a server configured to receive requests from a plurality of clients for content;*  
*an interface to a plurality of databases or data sources of content information*  
*coupled to said server;*  
*a communication protocol that provides a session connection between a client and*  
*the server, and allows the client to send, as part of the same session, a plurality of queries*  
*to query the server for content, wherein each of the plurality of queries are consecutive and*  
*form an increasingly focused query string for retrieving content from the server, and wherein*  
*each subsequent one of the plurality of queries extends the query string by one or more*  
*additional characters; and*  
*wherein said server simultaneously applies the increasingly focused query string*  
*against the plurality of databases or data sources as it is being extended, and suggests a*  
*set of increasingly appropriate content or search criteria from the plurality of databases, to*  
*the client, for further use by the client within the same session.*

Claim 52 defines a communication protocol that provides a session connection between a client and the server, and allows the client to send, as part of the same session, a plurality of queries to query the server for content. Each of the plurality of queries are consecutive and form an increasingly focused query string for retrieving content from the server. Each subsequent one of the plurality of queries extends the query string by one or more additional characters. The server simultaneously applies the increasingly focused query string against a plurality of databases or data sources as it is being extended, and suggests a set of increasingly appropriate content or search criteria from the plurality of databases, to the client, for further use by the client within the same session.

Yano discloses a document information management system in which a search-engine-compatible interface unit makes a word in a document displayed on the screen to be specified, transfers the specified word to a search engine as a keyword to be used in the search engine, receives a search result from the search engine, and displays the search result on the screen, while a browser-compatible interface unit performs a search (a keyword search and/or global search) by

using the keyword transferred from a browser and transfers a search result to the browser. (Abstract).

As further disclosed by way of example therein, at first the browser 102 displays a document on the screen (S701). When a character is identified (specified) through a mouse operation of the terminal unit 202 (S702), the search-engine-compatible interface unit 104 cuts out a character string obtained by adding some characters before and after the character thereto (adjacent character string) (S703), compares the cut-out character string to the word table (S704), cuts out the character string coincident with any word in the word table as a word, and identifies the word (S705). (Paragraphs [0075]-[0085]).

As shown in FIG. 8, for example, a cursor 801 is moved to an arbitrary character (herein a character "A" in the word "MARKED") in the document displayed on the screen, a mouse button is clicked, and the character (character "A") at the position of the cursor is specified. The character at the position of the cursor specified as described above is identified as an identified character 702A in step S702. Then, a character string obtained by adding some characters before and after the identified character thereto (adjacent character string) is cut out. In this case, the adjacent character string cut-out in step S703 is as follows: "when a plurality of buttons are marked a plurality of pages are outputted onto the display or to the printer." as indicated by the reference numeral 703A in FIG. 7. Then, the adjacent character string is compared with the words in the word table, and the character string coincident with any word in the word table is cut out as a word, and the word is identified. (Paragraphs [0075]-[0085]).

This identification of a word can be carried out as follows. 1) A character string "MA" obtained by adding one character before the specified character "A" is decided as a temporary word, and comparison is made as to whether there is any word coinciding with this temporary word in the word table or not, and when it is determined that there is a coinciding word in the word table, the temporary word is identified as a word. 2) When it is determined that there is no coinciding word in the word table, a character string "AR" obtained by adding one character behind the specified character "A" is decided as a temporary word, and comparison is made as to whether there is any word coinciding with this temporary word in the word table or not, and when it is determined that there is the coinciding word, the temporary word is identified as a word. 3) When it is determined that there is no coinciding word therein, a character string "MAR" obtained by adding one character

before and after the specified character "A" is decided as a temporary word, and comparison is made as to whether there is any word coinciding with this temporary word in the word table or not, and when it is determined that there is the coincident word therein, the temporary word is identified as a word. (Paragraphs [0075]-[0085]).

Thereafter and on, characters are added in the same manner as described above, and the processing is repeated until the cut-out character string is coincident with any word in the word table or until 'the cut-out character string' becomes the same as 'the adjacent character string'. Herein, a word "MARK" is identified as indicated by the reference numeral 705A in FIG. 7 during the identification of a word in step S705. The identified word "MARK" is recognized as a keyword in step S706 of FIG. 7, and the keyword is transferred to the search engine 103. The recognized keyword is also displayed on the screen in step S707. (Paragraphs [0075]-[0085]).

Applicant respectfully submits that, based on the above description, it appears that in Yano all of the query preparation steps are performed at the *browser*, particularly at the search-engine-compatible interface unit of the browser. The character string that is obtained by adding one character before the specified character, and the comparison that is made as to whether there is any word coinciding with this temporary word is performed by searching a word table, apparently at the browser. Subsequently, when it is determined that there is a coinciding word in the word table, the temporary word is identified as a word, again at the browser. When the cut-out character string is coincident with any word in the word table, or until the cut-out character string becomes the same as the adjacent character string, an identified word is recognized as a keyword. It appears only then is the keyword transferred from the browser to the search engine.

These steps also appear to be illustrated in the figures in Yano, in which a keyword is apparently prepared or recognized at the *browser* (e.g., S303, S603, S705) and is then sent only once, or as a *single request* to the search engine (e.g., S304, S605, S706). Yano does not appear to describe a keyword being sent from the browser to the search engine as a plurality of consecutive queries, or even as a plurality of queries.

Furthermore, as described above, Yano appears to be generally directed to a particular use case in that, for example, in the English language the words in a sentence are separated by a space, so that a character string sandwiched between spaces can without difficulty be considered as a word. However, in languages like Japanese, Chinese or Korean, words in a sentence are not

separated by space as in English. Therefore, in the documents written in these languages, it is difficult to decide where a word starts and where it ends in a sentence. In the above-described method, a character is added before or behind a specified character, so that, a stage is reached when a word having a meaning is formed and the word coincides with a word in the word table. This method is more effective for extracting a word from the documents written in Japanese, Chinese or Korean languages. (Paragraphs [0075]-[0085]).

As such, Applicant respectfully submits that Yano is directed to a different problem from embodiment defined by Claim 52. In particular, Yano is generally directed to determining the boundaries of words in a document that is displayed at a client browser, so that a keyword can be more clearly ascertained at that client browser, prior to sending that keyword to a search engine. Yano does not appear to suggest in any way sending portions of a word to a remote search engine or server. As such, Applicant respectfully submits that Yano does not disclose or render obvious the features of Claim 1, including that *the client sends, as part of the same session, a plurality of queries to query the server for content, wherein each of the plurality of queries are consecutive and form an increasingly focused query string* for retrieving content from the server, and wherein each subsequent one of the plurality of queries extends the query string by one or more additional characters; and wherein *said server simultaneously applies the increasingly focused query string against the plurality of databases or data sources as it is being extended*, and suggests a set of increasingly appropriate content or search criteria from the plurality of databases, to the client, for further use by the client within the same session.

In view of the above comments, Applicant respectfully submits that Claim 52 is neither anticipated by, nor obvious in view of the cited references, and reconsideration thereof is respectfully requested.

### **Claim 53**

The comments provided above with respect to Claim 52 are hereby incorporated by reference.

Applicant respectfully traverses the rejection of Claim 53, which defines similar features such as: providing a server configured to receive requests from a plurality of clients for content; providing access to a plurality of databases or data sources of content information coupled to said

server; providing a communication protocol that provides a session connection between a client and the server, and allows the client to send, as part of the same session, a plurality of queries to query the server for content, wherein each of the plurality of queries are consecutive and form an increasingly focused query string for retrieving content from the server, and wherein each subsequent one of the plurality of queries extends the query string by one or more additional characters; and simultaneously applying the increasingly focused query string against the plurality of databases or data sources as it is being extended, and suggests a set of increasingly appropriate content or search criteria from the plurality of databases, to the client, for further use by the client within the same session.

For similar reasons as provided above with respect to Claim 52, Applicant respectfully submits that Claim 53 is likewise neither anticipated by, nor obvious in view of the cited references, and reconsideration thereof is respectfully requested.

#### **V. Claim Rejections under 35 U.S.C. 103(a)**

In the Office Action mailed February 20, 2008, Claims 1-15, 17, 24-27, 29-51, 54-60 were rejected under 35 U.S.C. 103(a) as being unpatentable over Yano in view of Gershman (U.S. Patent No. 6,356,905).

#### **Claim 1**

The comments provided above with respect to Claim 52 are hereby incorporated by reference. Applicant respectfully traverses the rejection of Claim 1 for similar reasons as provided above. However, to expedite prosecution, Claim 1 has also been amended to more clearly define the embodiment therein as:

- 1. A system for retrieval at a client system of content from a server system, comprising: a communication protocol that enables an asynchronous connection over a network between a client system and a server system, and allows the client system to send via the network, and within a session between the client system and the server system, a lengthening string composed of a plurality of consecutively input characters, to query the server system for string-based content, while asynchronously receiving consecutive responses from the server as the characters are being input;*

*a client object, in communication with a client software at the client system and with the communication protocol, wherein the client object receives additional characters from the client software, and as consecutive characters are being received, transmits via the network to a server object at the server system a plurality of consecutive queries, within the session between the client system and the server system, to retrieve content from the server system, wherein each consecutive query lengthens the string by the additional characters, to form a lengthening string for retrieving matching content from the server system; and*

*a server object, in communication with the server system, and with the client object via the communication protocol, wherein the server object in response to receiving the consecutive queries that form the lengthening string, automatically uses the lengthening string to query and retrieve content information from the server system that matches the lengthening string, and wherein the server object asynchronously returns, while the additional characters are being input and the string is being lengthened during the session, consecutive responses containing increasingly matching content information to the client object for immediate use by the client system.*

As currently presented, Claim 1 defines a communication protocol that enables an asynchronous connection over a network between a client system and a server system, and allows the client system to send via the network, and within a session between the client system and the server system, a lengthening string composed of a plurality of consecutively input characters, to query the server system for string-based content, while receiving an asynchronous response from the server as the characters are being input.

Claim 1 further defines a client object, in communication with a client software at the client system and with the communication protocol, wherein the client object receives additional characters from the client software, and as consecutive characters are being received, transmits via the network to a server object at the server system a plurality of consecutive queries, within the session between the client system and the server system, to retrieve content from the server system, wherein each consecutive query lengthens the string by the additional characters, to form a lengthening string for retrieving matching content from the server system.

Claim 1 further defines a server object, in communication with the server system, and with the client object via the communication protocol, wherein the server object in response to receiving the consecutive queries that form the lengthening string, automatically uses the lengthening string to query and retrieve content information from the server system that matches the lengthening

string, and wherein the server object asynchronously returns, while the additional characters are being input and the string is being lengthened during the session, consecutive responses containing increasingly matching content information to the client object for immediate use by the client system.

As described above with regard to Claim 52, it appears that in Yano all of the query preparation steps are performed at the browser, and only then is the keyword transferred from the browser to the search engine. As further described above, Yano does not appear to describe a keyword being sent from the browser to the search engine as a plurality of consecutive queries, or even as a plurality of queries. As such, Applicant respectfully submits that Yano does not disclose or render obvious a number of features of Claim 1, including that the client object receives additional characters from the client software, and *as consecutive characters are being received, transmits via the network to a server object at the server system a plurality of consecutive queries*, within the session between the client system and the server system, to retrieve content from the server system, *wherein each consecutive query lengthens the string by the additional characters*, to form a lengthening string for retrieving matching content from the server system; and that the server object automatically uses the lengthening string to query and retrieve content information from the server system, and asynchronously returns, *while the additional characters are being input*, consecutive responses containing increasingly matching content information to the client.

In the Office Action mailed February 20, 2008, Yano was further characterized as disclosing at paragraphs [0035], [0045] and [0061] a server object, in communication with the server system, and with the client object via the communication protocol, wherein the server object in response to receiving the consecutive queries that form the lengthening string, automatically uses the lengthening string to query and retrieve content information from the server system that matches the lengthening string, and wherein the server object asynchronously returns, while the additional characters are being input and the string is being lengthened during the session, increasingly matching content information to the client object for immediate use by the client system.

However, Applicant respectfully submits that Yano in these sections appears to disclose that, when the user identifies a desired document out of a pop-up menu through a mouse operation the search-engine-compatible interface unit notifies the URL (identification of the document) of the corresponding document-described on the list to the browser. As shown in the corresponding



Figures 3 and 6, these steps (s306, s308, s609, s610) appear to be performed at the client. Only then does the browser access the database (Web server) with a document in the linked address stored therein according to the URL, and requests the database to transfer the source code of the corresponding document. As such, Applicant respectfully submits that Yano does not disclose that the server object asynchronously returns, *while the additional characters are being input and the string is being lengthened*, increasingly matching content information to the client.

Gershman was cited in the Office Action as disclosing the use of a communications protocol that enables an asynchronous connection over a network between a client system and a server system. Applicant respectfully submits that Gershman appear to disclose an Asynchronous Messaging component that can provide an asynchronous message based communication between the client and the server, using standard Internet mail protocols, SMTP and POP3s. (Column 48, lines 9-18). However, while SMTP and POP provide a form of email-based communication between a client and a server, Applicant respectfully submits that neither Gershman (nor Yano) disclose or render obvious the specific feature of a communication protocol that allows the client system to send a lengthening string composed of a plurality of consecutively input characters, to query the server system for string-based content, while asynchronously receiving consecutive responses from the server as the characters are being input. Claim 1 has also been amended as shown above to more clearly provide this distinction.

In view of the above comments, Applicant further respectfully submits that Claim 1 is neither anticipated by, nor obvious in view of the cited references, and reconsideration thereof is respectfully requested.

#### **Claim 24**

The comments provided above with respect to Claim 1 are hereby incorporated by reference. Applicant respectfully traverses the rejection of Claim 24. However, to expedite prosecution, Claim 24 has also been amended to more clearly define the embodiment therein as:

*24. A user interface mechanism, for use with a client application of a content retrieval system, said user interface mechanism indicating one or both of the availability of a session between said client application and a remote content server, and the status of said session, said mechanism comprising:*

*a user interface and input field, in communication with said client application, said input field allows a user to input data for transmission to a remote content server, wherein said input data includes a plurality of single string characters as part of a query;*

*a communication protocol that enables an asynchronous connection over a network between the client and the server, and allows the client to send, within a session between the client and the server, a plurality of consecutively input query strings via the network, to query the server for string-based content wherein the client receives additional characters from a user, and as each character is being received transmits to a server object at the server a plurality of consecutive queries, within the session between the client and the server, to retrieve content from the server, and while asynchronously receiving consecutive responses from the server as the characters are being input;*

*a server object, in communication with the server, and in communication with the client via the communication protocol, wherein the server object records, during the session, each of the plurality of consecutive queries from the client, and in response to receiving each query as it is being lengthened by one or more additional characters, automatically matches the lengthening query string against the content of the server, and asynchronously returns consecutive responses containing increasingly relevant content information to the client for immediate use by the client;*

*a session connection indicator, said session connection indicator displayed within a first portion of the input field, for indicating the availability of a connection between said client application and said content server; and*

*a status indicator, said status indicator displayed within the first or a second portion of the input field, for indicating during said session both the status of increasingly available content at said content server for selection by said user at that input field, and that the server object is currently using the lengthening query string against the content of the server system to query and retrieve content information from the server system.*

For similar reasons as provided above with respect to Claim 1, Applicant respectfully submits that Claim 24 is likewise neither anticipated by, nor obvious in view of the cited references. Additionally, Claim 24 further defines a *status indicator*, said status indicator displayed within the first or a second portion of the input field, *for indicating during said session both the status of increasingly available content at said content server for selection by said user at that input field, and that the server object is currently using the lengthening query string against the content of the server system to query and retrieve content information from the server system.* Applicant respectfully submits that this feature does not appear to be disclosed, anticipated by, or obvious in view of the cited references. Reconsideration is respectfully requested.

## Claim 29

The comments provided above with respect to Claim 1 are hereby incorporated by reference. Applicant respectfully traverses the rejection of Claim 29. However, to expedite prosecution, Claim 29 has also been amended to more clearly define the embodiment therein as:

29. *A method of providing communication at a client of string-based content from a server, comprising the steps of:*

*providing a communication protocol that enables an asynchronous session-based connection over a network between a client object and a server object, and allows the client object to send, within a session between the client object and the server object, a plurality of consecutively input query strings, to query the server for string-based content;*

*transmitting, via the client object in communication with said client, via the network to the server object a plurality of consecutive queries, within the session between the client object and the server object, to retrieve content from the server, wherein the client object receives additional characters from a user, and as each character is being received transmits to a server object at the server a plurality of consecutive queries, within the same session, to retrieve content from the server, wherein each consecutive query lengthens the query string by one or more characters, and forms a lengthening query string for retrieving content from the server; and*

*receiving, via said communication protocol, at the server object each of the plurality of consecutive queries from the client, and in response to receiving each query as it is being lengthened by one or more additional characters, automatically matching the lengthening query string against the content of the server, and asynchronously returning consecutive responses containing increasingly relevant content information to the client object for immediate use by the client.*

For similar reasons as provided above with respect to Claim 1, Applicant respectfully submits that Claim 29 is likewise neither anticipated by, nor obvious in view of the cited references. Reconsideration is respectfully requested.

## Claim 32

The comments provided above with respect to Claim 1 are hereby incorporated by reference. Applicant respectfully traverses the rejection of Claim 32, which currently defines:

32. *A system for session-based retrieval at a client of content from a server, comprising:*  
*a communication protocol that enables an asynchronous session over a network between a client and a server, and allows the client system to send, within a session between the client and the server, a plurality of consecutively input query strings, to query the server for content;*

*one or more content engine objects, in communication with the server object, that are capable of retrieving information from a content source containing string-based data by using a lengthening string as part of a content query and by returning matching data from the content source;*

*a user interface at the client that allows a user to enter a search string;*

*a client object, at the client, wherein the client object receives characters of the search string from the user interface as it is being entered by the user, and transmits via the network to a server object at the server a plurality of consecutive queries, within the same session, to retrieve content from the server system, wherein each consecutive query matches the characters of the search string as it is being entered, to form the lengthening search string for retrieving content from the server;*

*a server object, at the server, wherein the server object records, during the session, each of the plurality of consecutive queries from the client, and in response to receiving the lengthening search string from the client object, automatically matches the search string against the content of the server system, and asynchronously returns increasingly relevant content information to the client object for immediate use by the client; and*

*wherein the content information is used by the client to immediately update the user interface with options that match the content of the server system, as the user is entering the search string.*

For similar reasons as provided above with respect to Claim 1, Applicant respectfully submits that Claim 32 is likewise neither anticipated by, nor obvious in view of the cited references. Additionally, Claim 32 further defines one or more *content engine objects*, in communication with the server object, *that are capable of retrieving information from a content source containing string-based data by using a lengthening string as part of a content query and by returning matching data from the content source*, and wherein the content information is used by the client to immediately update the user interface with options that match the content of the server system, as the user is entering the search string. Applicant respectfully submits that this feature does not appear to be

disclosed, anticipated by, or obvious in view of the cited references. Reconsideration is respectfully requested.

### **Claim 33**

The comments provided above with respect to Claim 1 are hereby incorporated by reference. Applicant respectfully traverses the rejection of Claim 33, which currently defines:

33. *A method of providing session-based communication at a client of string-based content from a server, comprising the steps of:*

*providing a communication protocol that enables an asynchronous session over a network between a client and a server, and allows the client system to send, within a session between the client and the server, a plurality of consecutively input query strings, to query the server for content;*

*providing one or more content engine objects, in communication with the server object, that are capable of retrieving information from a content source containing string-based data by using a lengthening string as part of a content query and by returning matching data from the content source;*

*providing a user interface at the client that allows a user to enter a search string;*

*providing a client object, at the client, wherein the client object receives characters of the search string from the user interface as it is being entered by the user, and transmits via the network to a server object at the server a plurality of consecutive queries, within the same session, to retrieve content from the server system, wherein each consecutive query matches the characters of the search string as it is being entered, to form the lengthening search string for retrieving content from the server;*

*providing a server object, at the server, wherein the server object records, during the session, each of the plurality of consecutive queries from the client, and in response to receiving the lengthening search string from the client object, automatically matches the search string against the content of the server system, and asynchronously returns increasingly relevant content information to the client object for immediate use by the client; and*

*wherein the content information is used by the client to immediately update the user interface with options that match the content of the server system, as the user is entering the search string.*

For similar reasons as provided above with respect to Claim 1, Applicant respectfully submits that Claim 33 is likewise neither anticipated by, nor obvious in view of the cited references.

Additionally, Claim 33 further defines that *the content information is used by the client to immediately update the user interface with options that match the content of the server system, as the user is entering the search string.* Applicant respectfully submits that this feature does not appear to be disclosed, anticipated by, or obvious in view of the cited references. Reconsideration is respectfully requested.

### **Claim 36**

The comments provided above with respect to Claim 1 are hereby incorporated by reference. Applicant respectfully traverses the rejection of Claim 36, which currently defines:

36. *A system for providing session-based searching of string-based content from a server, comprising:*

*a user interface at a plurality of clients that allows a user at each of the plurality of clients to enter a string of consecutively input queries to query the server for string-based content, wherein each consecutive query lengthens the query string by one or more additional characters;*

*a communication protocol that transmits over a network, via a client object at each of said clients, to a server object at the server, the plurality of consecutive queries, to retrieve content from the server, wherein each additional character is immediately transmitted to the server object as the user is entering the additional characters in the user interface, to form an lengthening query string for retrieving content from the server; and*

*a server object which in response to receiving each query as it is being lengthened by the one or more additional characters, automatically matches the lengthening query string against the content of the server, and, as the user of a particular client is entering queries, asynchronously modifies the user interface by returning increasingly relevant server content information to the client object for immediate display to the user.*

For similar reasons as provided above with respect to Claim 1, Applicant respectfully submits that Claim 36 is likewise neither anticipated by, nor obvious in view of the cited references. Additionally, Claim 36 further defines that *as the user of a particular client is entering queries, the server object asynchronously modifies the user interface by returning increasingly relevant server content information to the client object for immediate display to the user.* Applicant respectfully submits that this feature does not appear to be disclosed, anticipated by, or obvious in view of the cited references. Reconsideration is respectfully requested.

### **Claim 37**

The comments provided above with respect to Claim 1 are hereby incorporated by reference. Applicant respectfully traverses the rejection of Claim 37, which currently defines:

*37. A method of providing session-based searching of string-based content from a server, comprising, comprising the steps of:*

*providing a user interface at a plurality of clients that allows a user at each of the plurality of clients to enter a string of consecutively input queries to query the server for string-based content, wherein each consecutive query lengthens the query string by one or more additional characters;*

*transmitting over a network, via a client object at each of said clients, to a server object at the server, the plurality of consecutive queries, to retrieve content from the server, wherein each additional character is immediately transmitted to the server object as the user is entering the additional characters in the user interface, to form an lengthening query string for retrieving content from the server; and*

*in response to receiving each query as it is being lengthened by the one or more additional characters, automatically matching the lengthening query string against the content of the server, and, as the user of a particular client is entering queries, asynchronously modifying the user interface by returning increasingly relevant server content information to the client object for immediate display to the user.*

For similar reasons as provided above with respect to Claim 1, Applicant respectfully submits that Claim 37 is likewise neither anticipated by, nor obvious in view of the cited references. Additionally, Claim 37 further defines that, in response to receiving each query as it is being lengthened by the one or more additional characters, the lengthening query string is automatically matched against the content of the server, and the user interface is asynchronously modified by returning increasingly relevant server content information to the client object. Applicant respectfully submits that this feature does not appear to be disclosed, anticipated by, or obvious in view of the cited references. Reconsideration is respectfully requested.

### **Claim 54**

The comments provided above with respect to Claim 1 are hereby incorporated by reference. Applicant respectfully traverses the rejection of Claim 54, which currently defines:

54. *A system comprising:*

*a client object on a client computer and a server object on a server computer, whereby the client computer and the server computer are linked by a network so that they can exchange information;*

*wherein the client object is linked to an input element in a user interface that allows a user to enter textual information comprising characters and strings to create incremental user input comprising a mutating string of characters;*

*wherein said user input is transmitted by the client object to the server object while said user input is being formed by a specific user during a user session;*

*wherein the server object uses said user input received from the client object to query data from one or more content sources, and to return result strings matching said user input asynchronously from said server computer while the input is being formed on the client computer; and*

*wherein the client object displays said results in a display element in the user interface on the client computer.*

For similar reasons as provided above with respect to Claim 1, Applicant respectfully submits that Claim 54 is likewise neither anticipated by, nor obvious in view of the cited references. Additionally, Claim 54 further defines that the client object is linked to an input element in a user interface that allows a user to enter textual information to create *incremental user input comprising a mutating string of characters*; and that user input is transmitted by the client object to the server object *while said user input is being formed*; and that the server object uses said user input received from the client object to query data from one or more content sources, and return result strings matching said user input asynchronously from said server computer while the input is being formed. Applicant respectfully submits that this feature does not appear to be disclosed, anticipated by, or obvious in view of the cited references. Reconsideration is respectfully requested.

**Claims 2-15, 17, 25-27, 30-31, 34-35, 38-51 and 55-60**

Claims 2-15, 17, 25-27, 30-31, 34-35, 38-51 and 55-60 depend from and include all of the features of one of Claims 1, 24, 29, 32, 33, 36, 37, 52, 53 or 54. Applicant respectfully submits that



each of these claims are allowable at least as depending from an allowable independent claim, and further in view of the comments provided above. Applicant further respectfully submits that many of the claims include features and limitations that may render the claims patentable in their own right; and respectfully reserves the right to argue these features and limitations should that be appropriate. However, to assist the Examiner in expediting prosecution, Applicant has provided comments below on several of the claims. Applicant welcomes the opportunity to further discuss the comments provided herein, if the Examiner feels such discussion would place the claims in better condition for allowance.

Claim 4 defines an embodiment in which the system comprises a plurality of server objects that run on a plurality of separate computers, and wherein said client queries are distributed over said separate computers. In the Office Action, it was submitted that Yano, at paragraph [0070], discloses such a feature. However, Applicant respectfully submits that Yano appears to disclose that preparation of the word table will be carried out as follows with terminal units 202, 203 and a Web server 201 connected to the network Net, which appears to describe the initial preparation of the word table. Yano does not appear to disclose *a plurality of server objects that run on a plurality of separate computers*, and wherein said client queries are distributed over said separate computers.

Claim 5 defines an embodiment in which server object stores previously received results from the server as stored results, and initially returns said stored results to the client in response to new client queries, without accessing the content at the server. In the Office Action, it was submitted that Yano, at paragraphs [0042] and [0058], discloses such a feature. However, Applicant respectfully submits that Yano appears to disclose that the search with the keyword in this case is performed based on the assumption that an object to be searched by the search engine (the database to be searched) is previously set and specified; and that a search can be performed across all the databases connected to the network, which appears to describe how words are previously set or specified, so that they can matched with a keyword. Yano does not appear to disclose that *the server object stores previously received results from the server as stored results, and initially returns said stored results to the client in response to new client queries*.

Claim 6 defines an embodiment in which client software is embedded into a software application that provides a visual interface that indicates to an operator that the server object is

currently using the lengthening query string against the content of the server system to query and retrieve content information from the server system and allows the operator to add additional characters to lengthen the query string, while simultaneously receiving and displaying increasingly matching results in consecutive responses from the server. In the Office Action, it was submitted that Yano, at paragraph [0053], [0065] and [0077], discloses such a feature. However, Applicant respectfully submits that Yano appears to disclose at these paragraphs the keyword determination that is being performed at the browser, prior to sending the keyword to the search engine. Yano does not appear to disclose *a visual interface that indicates to an operator that the server object is currently using the lengthening query string against the content of the server system to query and retrieve content information from the server.*

Claim 13 defines an embodiment in which client software identifies a user of the system to the server whereby the server can store statistics and provides a history of queries and corresponding responses appropriate to said user. In the Office Action, it was submitted that Yano, at paragraph [0125] and [0141], discloses such a feature. However, Applicant respectfully submits that while Yano appears to disclose a pop-up menu, Yano does not appear to disclose any means of *identifying a user of the system to the server, or wherein the server can store statistics and provides a history of queries and corresponding responses appropriate to said user.*

Claim 14 defines an embodiment in which server system comprises a server tier and a syndication tier, and wherein said client software communicates to the server tier on a single computer, and wherein each query is forwarded by the server tier and the syndication tier to an appropriate syndicate of content channels connected to the server tier on a different computer. In the Office Action, it was submitted that Yano, at paragraph [0137], discloses such a feature. However, Applicant respectfully submits that Yano does not appear to disclose the use of a syndication tier or content channels, wherein said client software communicates to the server tier on a single computer, and wherein each query is forwarded by the server tier and the syndication tier to an appropriate syndicate of content channels.

Claim 25 defines an embodiment in which several input fields in the user interface have session connection indicators and status indicators to indicate to the user the availability of a connection between said client application and said content server for those input fields, and the

status of increasingly available content at said content server for selection by said user at those input fields. In the Office Action, it was submitted that Yano, at paragraph [0073], disclose the feature of an application input field. However, Applicant respectfully submits that Yano does appear to disclose wherein *several input fields in the user interface have session connection indicators and status indicators* to indicate to the user the availability of a connection between said client application and said content server for those input fields.

Claim 30 defines an embodiment in which the server object matches each query received from the client against an in memory cache, and returns cached content to the client without accessing said content engine, unless the cached content has expired since it was last received from said content engine. In the Office Action, it was submitted that Yano, at paragraph [0105], discloses such a feature. However, Applicant respectfully submits that Yano appears to disclose a list display that is overwritten at the browser. Yano does not appear to disclose wherein *the server object matches each query received from the client* against an in memory cache, and *returns cached content* to the client without accessing said content engine.

Claim 31 defines an embodiment in which the server analyzes the time between said consecutive queries received from each client system, and skips selected ones of said consecutive queries to reduce network communications and the load on said content engine. In the Office Action, it was submitted that Yano, at paragraph [0075], discloses such a feature. However, Applicant respectfully submits that Yano appears to disclose that when a character is identified (specified) through a mouse operation of the terminal unit, the search-engine-compatible interface unit cuts out a character string obtained by adding some characters before and after the character thereto. Yano does not appear to disclose *analyzing the time between consecutive queries* received from each client system, and *skipping selected ones* of said consecutive queries.

Claim 38 defines an embodiment in which the client software is used to one of display suggestions, perform auto-completion, or provide type-ahead functionality, based on matching string-based data queried in a database by the server object on the server system. In the Office Action, it was submitted that Yano, at paragraph [0058], discloses such a feature. However, Applicant respectfully submits that Yano appears to disclose when the search engine receives a keyword from the search-engine-compatible interface unit it performs the processing of a search with the received keyword. Yano does not appear to disclose wherein *the client software is used* to

display suggestions, perform auto-completion, or provide type-ahead functionality, *based on matching string-based data queried in a database by the server object.*

Claim 58 defines an embodiment in which the input element on the client computer contains a visual object displayed within the display element that indicates to the user that user input was sent to the server object whereby the visual object keeps changing while matching results are being awaited from the server system, and whereby said visual object first changes when the user enters textual information, and before the user input is sent to the server, indicating to the user that the user input is being accumulated by the client object before sending it to the server object. In the Office Action, it was submitted that Yano, at paragraph [0152], discloses such a feature. However, Applicant respectfully submits that Yano appears to disclose a hypertext document that has a dummy button obtained by previously evaluating a word included in the document based on a word cut-out technology such as morpheme analysis. Yano does not appear to disclose *a visual object displayed within the display element that indicates to the user that user input was sent to the server object whereby the visual object keeps changing while matching results are being awaited from the server system.*

Claims 2-3, 7-12, 15, 17, 26-27, 34-35, 39-51, 55-57 and 59-60 are not addressed separately, but it is respectfully submitted that these claims are similarly allowable as depending from an allowable independent claim, and further in view of the comments provided above.

In view of the above comments, Applicant respectfully submits that Claims 2-15, 17, 25-27, 30-31, 34-35, 38-51 and 55-60 are neither anticipated by, nor obvious in view of the cited references, and reconsideration thereof is respectfully requested.

## **VI. Additional Amendments**

Claims 61-62 had been previously submitted in a Supplemental Reply file by Applicant on February 19, 2008. However, it appears the Supplemental Reply may have crossed with the mailing of the Office Action on February 20, 2008, and as such, Claims 61-62 may not have been entered in the Application. To reduce any likelihood of confusion, the present reply cancels Claims 61-62, and adds new Claims 63-64. Subject to the approval of the Examiner, Applicant respectfully requests that new Claims 63-64 be included in the Application and considered therewith.

**VII. Conclusion**

In view of the above amendments and remarks, it is respectfully submitted that all of the claims now pending in the subject patent application should be allowable, and reconsideration thereof is respectfully requested. The Examiner is respectfully requested to telephone the undersigned if he can assist in any way in expediting issuance of a patent.

Enclosed is a PETITION FOR EXTENSION OF TIME UNDER 37 C.F.R. §1.136 for extending the time to respond up to and including August 20, 2008.

The Commissioner is authorized to charge any underpayment or credit any overpayment to Deposit Account No. 06-1325 for any matter in connection with this response, including any fee for extension of time, which may be required.

Respectfully submitted,

Date: August 20, 2008

By: /Karl Kenna/  
Karl Kenna  
Reg. No. 45,445

Customer No.: 23910  
FLIESLER MEYER LLP  
650 California Street, 14<sup>th</sup> Floor  
San Francisco, California 94108  
Telephone: (415) 362-3800